

**REMARKS**

Claims 1-29 are pending in the application. Claims 1-7, 12-18 and 23-29 were rejected under 35 U.S.C. §102(a) as being anticipated by Kamel et al. Claims 8-11 and 19-22 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kamel as applied to claims 1-7, in view of Richter et al. Reconsideration and reexamination of the application in view of the following remarks is respectfully requested.

Claims 1-7, 12-18 and 23-29 were rejected under 35 U.S.C. §102(a) as being anticipated by Kamel. This rejection is respectfully traversed.

The present invention is directed to managing a shared read/write buffer pool and the execution of read and write commands to ensure that free blocks are available to temporarily store read or write data. A read command typically results in a number of pending read data requests that are satisfied over time, each of which requires the use of blocks in the shared read/write buffer pool. Without the management provided by the present invention, during the pendency of the read command, write commands may be initiated that consume the remaining buffer pool resources such that there will not be enough buffer pool resources to satisfy the pending read data requests. With both the pending read data requests and pending write commands consuming all of the buffer pool resources, neither may be able to complete, and a lockup condition may occur. To reduce the amount of read data re-transmissions, the write command may be throttled based upon the amount of pending read data requests that are currently unsatisfied and the amount of free blocks available. If the currently available free blocks would be substantially consumed by the total outstanding inbound read data requested, no more write commands will be initiated. As inbound read data is received into allocated buffers and transferred to the initiator device, the blocks in the buffer pool are freed up. When the read data transfer is completed or sufficient buffer resources have been freed up, the write data command may be initiated.

In particular, claims 1, 12 and 23 recite, in part: “*preventing an initiation of a new write data command until pending read data requests have been processed enough to free up sufficient blocks in the buffer pool to accommodate the data of the new write data command.*”

Kamel fails to disclose, teach or suggest these limitations. Kamel discloses a system in which all read and write requests are assigned deadlines, and the requests are placed in a single queue for processing in accordance with their priority (deadlines) (see col. 6 lines 31-52). Because all write requests are given deadlines and placed in the queue, no new write data command is ever prevented from being initiated (see col. 5 lines 27-34, col. 6 lines 47-49), as recited in claims 1, 12 and 23. Furthermore, the processing of pending read data requests in Kamel have no effect on the initiation of new write data commands, as recited in claims 1, 12 and 23. It should be understood that write requests have different characteristics from read requests and are handled differently from read requests (see col. 5 lines 18-20). Thus, a separate write buffer pool 28 is employed (see col. 5 lines 22-39), and the deadline assigned to each write request is the time  $d(t)$  before the write buffer pool gets full, which is based only on the rate  $\lambda_w$  at which write requests are received and the number of free buffer slots  $n_f(t)$  in the write buffer pool 28 (see col. 6 line 53 to col. 7 line 14). The calculated deadline of the write request determines when the write request is processed. Read requests have nothing to do with the calculated deadline (and therefore the priority) of the write requests. Kamel therefore actually teaches away from preventing an initiation of a new write data command until pending read data requests have been processed enough to free up sufficient blocks in the buffer pool, as recited in claims 1, 12 and 23.

Because Kamel does not disclose all of the limitations of claims 1, 12 and 23, the rejection of claims 1, 12 and 23 under 35 U.S.C. §102(a) as being anticipated by Kamel is respectfully traversed. In addition, because claims 2-7 depend from claim 1, claims 13-18 depend from claim 12, and claims 24-29 depend from claim 23, the rejection of those claims is traversed for the same reasons provided above with respect to claims 1, 12 and 23.

Claims 8-11 and 19-22 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kamel as applied to claims 1-7, in view of Richter. This rejection is respectfully traversed.

Claims 8-11 depend from claim 1, and claims 19-22 depend from claim 12. As discussed above, Kamel fails to disclose, teach or suggest preventing an initiation of a new write data command until pending read data requests have been processed enough to free up sufficient blocks

in the buffer pool, as recited in claims 1 and 12. Furthermore, Richter fails to make up for the deficiencies of Kamel with respect to claims 1 and 12, because like Kamel, Richter is completely silent as to “preventing an initiation of a new write data command until pending read requests have been processed enough to free up sufficient blocks in the buffer pool to accommodate the data of the new write data command,” as recited in claims 1, 12 and 23. In fact, Richter only discloses resource management in very general terms (see FIG. 5 and paragraphs [0211] through [0234].) Richter mentions that when a request for content is received (see paragraph [0214]), the requirements for fulfilling that request may be determined (see paragraph [0216]), an indication of whether there are enough available resources may be generated (see paragraph [0219]), and a system monitor may even be told to temporarily slow down on accepting requests for content until the necessary resources become available (see paragraph [0221]). However, there is no disclosure at all in Richter related to write commands.

Therefore, even if it is assumed that the requisite motivation exists to combine Kamel and Richter, the limitation “preventing an initiation of a new write data command until pending read requests have been processed enough to free up sufficient blocks in the buffer pool to accommodate the data of the new write data command” is completely lacking in both Kamel and Richter.

Because neither Kamel nor Richter, alone or in combination, discloses, teaches or suggests all of the limitations of amended claims 1 and 12, and because claims 8-11 depend from claim 1, and claims 19-22 depend from claim 12, it is respectfully submitted that the rejection of claims 8-11 and 19-22 under 35 U.S.C. §103(a) as being unpatentable over Kamel as applied to claims 1-7, in view of Richter, has been overcome.

In view of the above, each of the presently pending claims in this application is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to pass this application to issue.

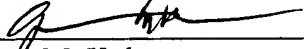
If, for any reason, the Examiner finds the application other than in condition for allowance, Applicants request that the Examiner contact the undersigned attorney at the Los Angeles

telephone number (213) 892-5752 to discuss any steps necessary to place the application in condition for allowance.

In the unlikely event that the transmittal letter is separated from this document and the Patent Office determines that an extension and/or other relief is required, Applicants petition for any required relief including extensions of time and authorizes the Commissioner to charge the cost of such petitions and/or other fees due in connection with the filing of this document to **Deposit Account No. 03-1952** referencing Docket No. 491442001600.

Dated: November 28, 2005

Respectfully submitted,

By   
Glenn M. Kubota  
Registration No.: 44,197  
MORRISON & FOERSTER LLP  
555 West Fifth Street, Suite 3500  
Los Angeles, California 90013  
(213) 892-5200